Claims

1. A computer-implemented process for identifying security vulnerabilities in a host computer system via a scanner, including the steps of:

passing exploit objects, which contain exploits that check a host computer system for vulnerabilities and resource objects, which contain resources which can be used by the scanner, from an exploit manager and a resource manager to an engine of the scanner; and

executing exploits, which check a host computer system for vulnerabilities, contained in the exploit objects via the engine to identify security vulnerabilities in the host computer system.

2. A computer-implemented process for identifying security vulnerabilities in a host computer system via a scanner, including the steps of:

installing an express update package containing: an exploit plug-in module containing exploit objects, which contain exploits that check a host computer system for vulnerabilities; a resource plug-in module containing resource objects, which contain resources which can be used by the scanner; a dat file, which contains exploit attribute information; and a help file, which contains on-line help information, on a computer;

supplying exploit attribute information to an exploit manager from a dat file;

passing exploit object and resource object information from the exploit manager and the resource manager to an engine of the scanner; and

executing exploits.

3. The computer-implemented process of claim 2 wherein said resources can be assigned a namespace based upon the resource's scope.

4. The computer-implemented process of claim 2, wherein said step of executing exploits includes the steps of:

running standard built-in exploits;

running standard plug-in exploits;

running denial of service plug-in exploits; and

running denial of service built-in exploits.

5. The computer-implemented process of claim 4, wherein said steps of running standard and denial of service built-in exploits includes the steps of:

having the engine get the exploit at the top of a run-order list;

having the engine attempt to run the exploit;

if the exploit is run, recording the exploit result information to a database and a scanner log file;

sending the exploit result information to a user interface to display; and

repeating the above steps for the remaining exploits.

6. The computer-implemented process of claim 4, wherein said steps of running standard and denial of service plug-in exploits includes the steps of:

having the plug-in engine make copies of the master exploit list (a list of exploits and the resources the exploits produce and consume) and the master resource list (a list of resources and the exploits that produce and consume those resources) from the session object;

getting exploit information from the scanpolicy object for the first exploit;

creating a target object and putting the exploit information in the target object;

passing the target object to the exploit object;

running the exploit;

adding exploit result information to the target object;

passing the target object back to a plug-in engine;

querying the target object for exploit result information;

recording exploit result information to the scanner log file and sending the exploit result information to the user interface; and

repeating the above steps for the remaining exploits.

7. The computer-implemented process of claim 6, wherein said step of repeating the above steps for the remaining exploits includes the steps of:

running exploits that neither produce nor consume shared resources;

running exploits that only produce shared resources;

running exploits that produce and consume shared resources; and

running exploits that only consume shared resources.

8. The computer-implemented process of claim 7, wherein said step of running exploits that produce and consume shared resources includes the step of ensuring that all producers of resources consumed by the exploit are run before the exploit is run.

9. The computer-implemented process of claim 2, further including the step of initializing a scanner.

10. The computer-implemented process of 9, wherein the step of initializing a scanner includes the steps of

enumerating plug-in modules and objects;

running load security for each plug-in module;

initializing a policy manager.

11. The computer-implemented process of 10, wherein the step of initializing a policy manager includes the steps of:

asking an exploit manager and a resource manager to identify available exploits and resources;

having the exploit manager and the resource manager query the registry for available exploit objects and the available resource objects;

having the exploit manager and the resource manager create maps indicating which plug-in modules contain the available exploit objects and the available resource objects;

having a policy manager ask the exploit manager and resource manager for the available exploit objects and common-setting resource objects;

creating the available exploit objects and common-setting resource objects;

having the policy manager query the available exploit objects and common-setting resource objects.

12. The computer-implemented process of claim 2, including the step of getting license, policy and host information.

13. The computer-implemented process of claim 12, wherein said step of getting policy information includes the steps of:

having the policy manager create a scanpolicy object;

having a policy editor examine, modify and configure exploit and resource policy settings; and

having the policy editor store the user choices in a policy file.

14. The computer-implemented process of claim 13, wherein said step of having a policy manager create the scanpolicy object includes the steps of:

having the policy editor query the policy manager for policy information about an exploit; and

having the policy manager give the policy information to the policy editor.

15. The computer-implemented process of 2, including the steps of:

having host-scanning threads query a session manager for available hosts to scan;

having the session manager query the session objects for the next host; and

having the session manager return the host to the host-scanning thread.

16. The computer-implemented process of claim 2, wherein the step of running security checks is included.